

Decidability of Equivalence of Aggregate Count-Distinct Queries

Babak Bagheri Harari Val Tannen
Computer & Information Science Department
University of Pennsylvania

ABSTRACT

We address the problem of equivalence of count-distinct aggregate queries, prove that the problem is decidable, and can be decided in the third level of Polynomial hierarchy. We introduce the notion of core for conjunctive queries with comparisons as an extension of the classical notion for relational queries, and prove that the existence of isomorphism among cores of queries is a sufficient and necessary condition for equivalence of conjunctive queries with comparisons similar to the classical relational setting. However, it is not a necessary condition for equivalence of count-distinct queries. We introduce a relaxation of this condition based on a new notion, which is a potentially new query equivalent to the initial query, introduced to capture the behavior of count-distinct operator.

1. INTRODUCTION

The problem of deciding equivalence among conjunctive aggregate count-distinct queries has been investigated starting from late 90s, and the problem is known to be decidable for most of the aggregate operators [3, 5, 1, 2, 5] including SUM, Average, and COUNT. For count-distinct queries, although some sufficient conditions have been proposed [5, 2], it has still been open if the problem is decidable or not. In this paper we study the problem of equality of count-distinct conjunctive queries, and provide a sufficient and necessary condition for the problem. First, we introduce the notion of core for conjunctive queries with comparisons. We show that similar to the classic results for relational conjunctive queries, the existence of an isomorphism between cores of conjunctive queries with comparisons is a sufficient and necessary conditions for their equality. While this condition provides us with a sufficient condition for the equality of conjunctive queries, it is easy to show that existence of isomorphism is not a necessary condition for equivalence of count-distinct queries. The necessary and sufficient condition is obtained by a relaxation of the above condition based on a newly introduced notion of flip of a query. The flip of a count-distinct query is a potentially different count-distinct

query equal to the original query, obtained by flipping the direction of some of the comparisons among the variables of the query. We show that given two queries to compare, they are equal if and only if the first query or its flip is isomorphic to the second query or its flip. Both computing of the core of a query and its flip can be done in Δ_3^π , which makes an upper-bound for the complexity of our suggested algorithm.

2. PRELIMINARIES

We consider an ordered dense domain Δ , and a countably infinite set \mathcal{V} of *variables*. *Terms* \mathcal{T} are either constants from Δ or variables, i.e., $\mathcal{T} = \Delta \cup \mathcal{V}$.

Relations. Given n possibly different sets $T_1, \dots, T_n \subseteq \mathcal{T}$ of terms, an n -ary relation R/n between them is a subset of their cartesian products: $R \subseteq T_1 \times \dots \times T_n$. We might simply use the term *relation*, when the arity is implicit or not relevant. The *identity relation* over a given set $T \subseteq \mathcal{T}$, denoted as $id(T)$ is the bijection relation from T to T that maps all the terms to themselves, i.e., $id(t) = t$ for all $t \in T$. Given a binary relation $R : S \rightarrow S'$, we denote with $\text{DOM}(R)$ the domain of R , i.e., the set of elements in S on which R is defined. We denote with $\text{IM}(R)$ the set of elements s' in S' such that $(s, s') \in R$ for some $s \in S$. A *function (mapping)* $f : T_1 \mapsto T_2$ is a binary relation between T_1 and T_2 such that for all terms $\{a, b, c\} \subseteq \mathcal{T}$ such that $(a, b) \in f$ and $(a, c) \in f$, $b = c$. For functions, we might use $f(a) = b$, instead of $(a, b) \in f$. A *bijection* h between two sets T_1 and T_2 is a function from T_1 to T_2 , such that for all $a \in T_1$ there is exactly one $b \in T_2$ such that $h(a) = b$, and for all $b \in T_2$ there is exactly one $a \in T_1$ such that $h(a) = b$. A *partial bijection* between two sets \mathcal{T}_1 and \mathcal{T}_2 is a bijection between a subset of \mathcal{T}_1 and a subset of \mathcal{T}_2 . Given a bijection h , the inverse of h denoted as h^{-1} is the function for which for all pairs a and b , $(a, b) \in h$ if and only if $(b, a) \in h^{-1}$. Given a binary relation R and a set S , the *restriction* of R to S , denoted as $R|_S$ is the relation obtained by restricting the domain of R to the elements of S as follows: $\{(t_1, t_2) \mid (t_1, t_2) \in R, \text{ and } t_1 \in S\}$. We say a function f' *extends* the function f , if f' is a function, and $f \subseteq f'$.

Databases. We assume a countably infinite set \mathcal{R} of *relation names*, and to each relation name $R \in \mathcal{R}$ assign an arity $\text{ARITY}(R)$ greater than or equal to zero. A *relational schema* is a finite set of relation names with specified arities,

i.e., is a finite subset of \mathcal{R} . Given a relation name $R \in \mathcal{R}$, an instance of R over the set $I \subseteq \mathcal{T}$ is a finite subset of $I^{\text{arity}(R)}$. Given a relational schema $S = \{R_1, \dots, R_n\}$, a *relational instance* of S over $I \subseteq \mathcal{T}$ is a set of instances of R_1, \dots, R_n . Given a database instance \mathcal{I} , its *active domain* $\text{adom}(\mathcal{I})$ is the subset of \mathcal{T} such that $u \in \text{adom}(\mathcal{I})$ if and only if u occurs in \mathcal{I} .

Assignment. An *assignment* is a partial function $\theta : \mathcal{T} \mapsto \mathcal{T}$, such as $\{t_1 \rightarrow c_1, \dots, t_n \rightarrow c_n\}$ which assigns the constants $c_1, \dots, c_n \subseteq \Delta$ to the terms t_1, \dots, t_n . Given a relational instance I and an assignment θ , we use the notations $I\theta$ and $\theta(I)$ to denote the instance obtained by applying the substitution θ to the terms of I . Formally,

$$I\theta = \theta(I) = \{ R(c_1 \dots c_n) \mid R(t_1, \dots, t_n) \in I, \text{ and } (t_i \rightarrow c_i) \in \theta \text{ for } i \in \{1, \dots, n\} \}$$

Homomorphism and Isomorphism Given a bijection σ , and two relational instances I and J , a σ -homomorphism h from I to J is a mapping from the $\text{adom}(I)$ to $\text{adom}(J)$, defined as follows:

1. h is a function $\text{adom}(I) \mapsto \text{adom}(J)$,
2. h extends σ , i.e., $\sigma \subseteq h$,
3. for all relations names R with arity n , $R(a_1, \dots, a_n) \in I$ implies $R(h(a_1), \dots, h(a_n)) \in J$.

We say that there is a homomorphism from I to J wrt the bijection σ , denoted as $I \triangleright^\sigma J$ in case such a homomorphism relation exists. We refer to h as a witness of the homomorphism from I to J . We might use the notation $I \triangleright_h^\sigma J$ when needed to emphasize that h is a witness of the homomorphism from I to J . In case σ is the bijection relation $\text{id}(C)$, we might simply use C to refer to the bijection relation σ . When it is non-relevant or clear from the context, we might drop σ , and simply say there is a homomorphism from I to J , denoted as $I \triangleright J$.

We say that I is σ -isomorphic to J , denoted as $I \equiv^\sigma J$ if there exists a bijection relation h , such that $I \triangleright_h^\sigma J$ and $J \triangleright_{h^{-1}}^\sigma I$. When it is clear from the context, we might drop σ , and simply say I and J are isomorphic, denoted as $I \equiv J$. We also say that h is a witness of the σ -isomorphism from I to J , and when it is clear from the context we might drop σ , and simply say that h is a witness of the isomorphism from I to J .

Conjunctive Queries with Comparisons. Here we borrow the definition of [4] with following naming conventions. A q over the schema \mathcal{S} can be specified with (i) a set $\text{disting}(q) = \{x_1, \dots, x_m\}$ of distinguished variables, the sequence being called the *summary row*, or just the summary; (ii) A set $\text{nondisting}(q) = \{y_1, \dots, y_n\}$ of *non-distinguished* (existentially quantified) variables; (iii) A set $\text{rel}(q) = c_1, \dots, c_l$ of distinct conjuncts, each conjunct c_i being an atomic formula of the form $R(z_1, \dots, z_r)$, where $R/r \in \mathcal{S}$ and each z_i is

a variable or a constant, i.e., $z_i \in \text{disting}(q) \cup \text{nondisting}(q) \cup \Delta$; (iv) A set $\text{comp}(q) = \{l_1, \dots, l_u\}$ of comparisons in the form $z_i \sigma z_j$ in which $z_i, z_j \in \text{disting}(q) \cup \text{nondisting}(q) \cup \Delta$, and $\sigma \in \{=, <, \leq\}$. We denote the set of all distinguished and non-distinguished variables of q with $\text{var}(q)$, i.e., $\text{var}(q) = \text{disting}(q) \cup \text{nondisting}(q)$.

Given a relational instance I , and a conjunctive query q , the answer of q over I , denoted as $\text{ans}(q, I)$ is a set of assignments in the form $\sigma : \text{nondisting}(q) \mapsto \text{adom}(I)$ such that for each of the assignment σ there exists an assignment $\sigma' : \text{disting}(q) \mapsto \text{adom}(I)$, and for $\sigma'' = \sigma \cup \sigma'$ followings hold:

1. for each $c_i \in \text{rel}(q)$, $\sigma''(c_i) \in I$
2. for each $l_i = z_i \theta z_j \in \text{comp}(q)$, $\sigma''(z_i) \theta \sigma''(z_j)$.

Given a query ϕ and an assignment θ , we use the notation $\phi\theta$ to denote the query obtained by applying θ to the variables of ϕ .

Homomorphism and Isomorphism of Conjunctive Queries with Comparisons.

Given a bijection σ , and queries $q_1(\bar{x}, c(y)) :- A_1(\bar{x}, y, z) \wedge C_1(\bar{x}, y, z)$, and $q_2(\bar{x}, c(y)) :- A_2(\bar{x}, y, z) \wedge C_2(\bar{x}, y, z)$, a σ -homomorphism h from q_1 to q_2 is a mapping such that $A_1 \triangleright_h^{\sigma'} A_2$ for $\sigma' = \sigma \cup (\bar{x} \mapsto \bar{x})$, and for all linearization L of C_2 , for all comparison $(r \sigma s) \in C_1$, $L \models h(r) \sigma h(s)$.

We say that there is a homomorphism from q_1 to q_2 wrt the bijection σ , denoted as $q_1 \triangleright^\sigma q_2$ in case such a homomorphism relation exists. We refer to h as a witness of the homomorphism from q_1 to q_2 . We might use the notation $q_1 \triangleright_h^\sigma q_2$ when needed to emphasize that h is a witness of the homomorphism from q_1 to q_2 . In case σ is the bijection relation $\text{id}(C)$, we might simply use C to refer to the bijection relation σ . When it is non-relevant or clear from the context, we might drop σ , and simply say there is a homomorphism from q_1 to q_2 , denoted as $q_1 \triangleright q_2$.

We say that q_1 is σ -isomorphic to q_2 , denoted as $q_1 \equiv^\sigma q_2$ if there exists a bijection relation h , such that:

1. h extends σ , i.e., $\sigma \subseteq h$,
2. for all relations names R with arity n , $R(a_1, \dots, a_n) \in I$ implies $R(h(a_1), \dots, h(a_n)) \in J$.
3. for all relations names R with arity n , $R(a_1, \dots, a_n) \in J$ implies $R(h^{-1}(a_1), \dots, h^{-1}(a_n)) \in I$;
4. for all comparisons $r \sigma s \in C_1$, $h(r) \sigma h(s) \in C_2$;
5. for all comparisons $r \sigma s \in C_2$, $h^{-1}(s) \in C_1$.

When it is clear from the context, we might drop σ , and simply say q_1 and q_2 are isomorphic, denoted as $q_1 \equiv q_2$. We also say that h is a witness of the σ -isomorphism from q_1 to q_2 , and when it is clear from the context we might drop σ ,

and simply say that h is a witness of the isomorphism from q_1 to q_2 .

Extension of a Query. Given a count-distinct query $q(\bar{x}, c(y)) :- A(\bar{x}, y, z) \wedge C(\bar{x}, y, z)$ the extension of q denoted as $\text{ext}(q)$ is the count distinct query $q'(\bar{x}, c(y)) :- A(\bar{x}, y, z) \wedge C'(\bar{x}, y, z)$, in which C' is a set of comparisons defined as follows:

$$C' = \{r \sigma s \mid r, s \in \text{atom}(q) \text{ and } C \models r \sigma s\}.$$

EXAMPLE 2.1. Lets assume $q(\bar{x}, c(y)) :- A(x) \wedge A(y) \wedge B(z) \wedge x < y \wedge y < z$. Then the extension of q is $q(\bar{x}, c(y)) :- A(x) \wedge A(y) \wedge B(z) \wedge x < y \wedge y < z \wedge x < z$.

It is easy to see that the extension of a query always exists and is homomorphically equivalent to the original query.

3. EQUIVALENCE OF COUNT-DISTINCT QUERIES

In this section we address our main results on equivalence of count-distinct queries. We first introduce the notion of core of conjunctive queries with comparison, and show that existence of isomorphism among core of queries coincides with their equality. Then we will introduce a relaxation of this condition through the notion of flip of conjunctive queries, which is introduced in this paper to capture when two non-isomorphic count-distinct queries can be equal. Finally, we will prove that two count-distinct queries are equal if and only if at least one of the pairs of the original query or their flips are isomorphic.

3.1 Equivalence of Conjunctive Queries with Comparison

Here we introduce the core of conjunctive queries with comparisons as a generalization of the concept of core for relational conjunctive queries. We will show that our notion of core coincides with the classic definition when restricted to the relational queries. Moreover, similar to the classic notion of core, existence of isomorphism among core of queries with comparisons implies equality of the queries, and vice versa.

DEFINITION 3.1 (CORE). Given a conjunctive query with comparisons $q(\bar{x}, c(y)) :- A(\bar{x}, y, z) \wedge C(\bar{x}, y, z)$, a core of q is a query such as $q'(\bar{x}, c(y)) :- A'(\bar{x}, y, z) \wedge C'(\bar{x}, y, z)$ such that:

1. There is a homomorphism from q to q' , i.e., $q \triangleright^{\bar{x}} q'$;
2. for all queries q'' that is homomorphically equivalent to q , there is an injective homomorphism from q' to q'' , i.e.,

$$q \triangleright^{\bar{x}} q'', q'' \triangleright^{\bar{x}} q \implies q' \triangleright_h^{\bar{x}} q''$$

in which h is an injective function;

3. $q' = \text{ext}(q')$.

Following theorem shows that core of count-distinct queries is unique up to isomorphically equivalence.

THEOREM 3.2. Given a conjunctive query with comparisons $q(\bar{x}, c(y)) :- A(\bar{x}, y, z) \wedge C(\bar{x}, y, z)$ if q_1 and q_2 are cores of q then $q_1 \equiv^{\bar{x}} q_2$.

EXAMPLE 3.3. Consider the following query.

$q(\bar{x}, c(y)) :- A(x) \wedge A(y) \wedge A(z) \wedge A(t) \wedge x < y \wedge y < z \wedge y < t$
core(q) is

$$q(\bar{x}, c(y)) :- A(x) \wedge A(y) \wedge A(z) \wedge x < y \wedge y < z \wedge x < z$$

LEMMA 3.4. Core of conjunctive queries with comparisons can be computed in Δ_3^P .

LEMMA 3.5. The notion of core of conjunctive queries with comparisons is equivalent to the classic notion of core conjunctive queries for relational queries.

LEMMA 3.6. Two conjunctive queries with comparisons are equal if and only if their cores are isomorphic.

LEMMA 3.7. If the core of two count-distinct queries are isomorphic then they are equivalent.

3.2 Equivalence of Count-Distinct Aggregate Queries

3.2.1 Flip of a Query

Isomorphism of cores of queries is not a necessary condition for equality of count-distinct queries, and to find a sufficient and necessary condition we introduce a relaxation of this condition. First, we introduce the notion of flip of a query which is a potentially new count-distinct query. We show that existence of isomorphism between a query or its flip with another query or its flip is a sufficient and necessary condition for their equivalence. In order to define the flip of a query, first we introduce the notion of equivalence set of a variable in a query as a subset of the variables of the query containing the initial variable such that all the variables in the set are homomorphically equivalent to each other forgetting the comparisons among them.

DEFINITION 3.8 (EQUAL SET OF A VARIABLE). Given a count-distinct query $q(\bar{x}, c(y)) :- A(\bar{x}, y, z) \wedge C(\bar{x}, y, z)$, and a variable $v \in \text{var}(q)$, a potential equal set of v w.r.t. q is a set $S \subseteq \text{var}(q)$ such that for all variables in S are homomorphically equivalent in the query obtained from dropping the comparisons among those of S from q . More specifically, lets consider q' to be the query obtained from q by dropping all comparisons among the variables in S . For all variables x_1, x_2 in S :

$$\begin{cases} q' \triangleright^{\sigma} q', & \text{for } \sigma = \bar{x}x_1 \mapsto \bar{x}x_2; \\ q' \triangleright^{\sigma'} q', & \text{for } \sigma' = \bar{x}x_2 \mapsto \bar{x}x_1. \end{cases}$$

Notice that for each variable v , $\{v\}$ is a potential equal set. Moreover, all the equal sets of a variable are subsets of the variables of the query. Now in Lemmas 3.9, 3.10, and 3.11, and Theorem 3.12 we prove that each variable has a unique biggest equal set.

LEMMA 3.9. *Lets consider C to be a satisfiable set of comparisons without equality. Followings hold:*

1. *For all linearization L of C , if $L \models m \leq n$ then there exists a set of terms $\{r_1, \dots, r_m\}$ such that*

$$(m \ \rho_1 \ r_1 \ \rho_2 \ \dots \ \rho_{m-1} \ r_m \ \rho_m \ n) \in C$$

for $\rho \in \{<, \leq\}$. Moreover, if $L \models m < n$, then ρ_i is strict $<$ for at least one $i \in \{1, \dots, m\}$.

2. *If there exists a set of terms $\{r_1, \dots, r_m\}$ such that*

$$(m \ \rho_1 \ r_1 \ \rho_2 \ \dots \ \rho_{m-1} \ r_m \ \rho_m \ n) \in C$$

for $\rho \in \{<, \leq\}$, then for all linearization L of C , $L \models m \leq n$. Moreover, if ρ_i is strict $<$ for at least one $i \in \{1, \dots, m\}$, then for all linearization L of C , $L \models m < n$.

LEMMA 3.10. *Lets consider the count-distinct query $q(\bar{x}, c(y)) :- A(\bar{x}, y, z) \wedge C(\bar{x}, y, z)$, a variable $v \in \text{var}(q)$, M to be a partial equal set of v w.r.t. q , $x, y \in M$, and C_M to be the set of comparisons obtained from C by removing those among variables in M . If for all linearization L of $C(\bar{x}, y, z)$, $L \models x \rho r$ for some variable $r \notin M$, then for all linearization L_M of C_M , $L_M \models y \rho r$.*

PROOF. Since for all linearization L of C $L \models x \rho r$, by Lemma 3.9 there exists a chain of comparisons $x \rho_1 x_1 \rho_2 \dots \rho_m r \in C$. Lets assume x_i is the last element in this chain that is the member of M . Considering the fact there is no variable from M in the path from x_i to r , and using Lemma 3.9, for all linearization L_M of C_M , $L_M \models x_i \rho r$. Finally, given the fact that x_i and y are both in M , and by definition they are homomorphically equivalent in the query obtained by dropping the comparisons among variables of M , $L_M \models y \rho r$. \square

LEMMA 3.11. *Lets consider a count-distinct query $q(\bar{x}, c(y)) :- A(\bar{x}, y, z) \wedge C(\bar{x}, y, z)$, a variable $v \in \text{var}(q)$, and an equal set M of v w.r.t. q . For variables x and y in q , if there exists a path*

$$p : x \ \rho_1 \ x_1 \ \dots \ \rho_m \ x_m \ \rho_{m+1} y$$

in $C(\bar{x}, y, z)$, then there exists a path

$$x \ \rho'_1 \ x'_1 \ \dots \ \rho'_n \ x'_n \ \rho'_{m+1} y$$

in $C(\bar{x}, y, z)$ that does not contain any comparison among the variables in M .

PROOF. Consider a sub-path $x_i \rho_i x_{i+1} \dots x_k \rho n$ of p for which $\{x_i, \dots, x_k\}$ to M . Using Lemma 3.10, we can replace such a path by another path between x_i and n such that it does not contain any comparison among variables of

M . Consequently, from p , we can obtain a path $x \ \rho'_1 \ x'_1 \ \dots \ \rho'_n \ x'_n \ \rho'_{m+1} y$ in $C(\bar{x}, y, z)$ that does not contain any comparison among the variables in M . \square

Next Theorem shows that there exists a unique maximal equal set for each variable of a given count-distinct query, partition those variables. In the rest of the paper, we refer to the biggest equal set to which a variable belongs wrt q as $\text{equiv}(x, q)$.

THEOREM 3.12. *Given a count-distinct query, if a variable has two different equal sets, then one of them is a subset of the other.*

PROOF. Lets assume there are two different equal sets M and N to which a variable n belongs. We will show that the set $U = M \cup N$ also satisfies all conditions of an equal set, so contradicts the assumption that M and N are maximal sets satisfying the conditions of equal sets.

Lets consider the query $q(\bar{x}, \text{contd}(y)) :- A(\bar{x}, y, z) \wedge C(\bar{x}, y, z)$, and $q_U :- A(\bar{x}, y, z) \wedge C_U(\bar{x}, y, z)$ to be the query obtained from q by dropping all comparisons among the variables in U . Lets consider two variables $x_1, x_2 \in U$. We need to show that x_1 and x_2 are homomorphically equivalent wrt q_U : i.e., $q_U \xrightarrow{\sigma} q_U$, for $\sigma = \bar{x}x_1 \mapsto \bar{x}x_2$ and $q_U \xrightarrow{\sigma'} q_U$, for $\sigma' = \bar{x}x_2 \mapsto \bar{x}x_1$.

One of the followings hold:

- 1- $x_1, x_2 \in M$. Lets consider $q_M :- A(\bar{x}, y, z) \wedge C_M(\bar{x}, y, z)$ to be the query obtained from q by dropping all comparisons among the variables in M . Since x_1 and x_2 are in M , by definition we know that $q_M \xrightarrow{\sigma} q_M$, for $\sigma = \bar{x}x_1 \mapsto \bar{x}x_2$. That means there exist a function h that extends σ and is satisfying the conditions of the homomorphism from q_M to q_M , i.e.,:

- For all $R(r_1, \dots, r_m) \in q_M$, $R(h(r_1), \dots, h(r_m)) \in q_M$;
- For all linearization L_M of $C_M(\bar{x}, y, \bar{z})$, for all comparisons $r \rho s \in C_M(\bar{x}, y, \bar{z})$, $L_M \models h(r) \rho h(s)$, for $\rho \in \{<, >, \leq, \geq\}$.

Now we show that h , is also a witness of the homomorphism $q_U \xrightarrow{\sigma} q_U$. Since q_M and q_U only differ on the comparisons among the variables, the first condition of the homomorphism holds, i.e., for all $R(r_1, \dots, r_m) \in q_U$, $R(h(r_1), \dots, h(r_m)) \in q_U$. So we only need to show that for all linearization L_U of $C_U(\bar{x}, y, \bar{z})$, for all comparisons $r \rho s \in C_U(\bar{x}, y, \bar{z})$, $L_U \models h(r) \rho h(s)$. Lets assume $\rho = \leq$. For other cases we can prove similarly. Since for all linearization L_M of C_M , $L_M \models h(r) \rho h(s)$, using Lemma 3.9, we know that there exists following chain for comparisons $h(r) \rho_1 r_1 \rho_2 r_2 \dots \rho_m r_m \rho_{m+1} h(s)$ in C_M , such that $\rho_i \in \{<, \leq\}$. Moreover, using Lemma 3.11, we know that there exists a path $x \ \rho'_1 \ x'_1 \ \dots \ \rho'_n \ x'_n \ \rho'_{m+1} y$ in $C(\bar{x}, y, z)$ that does not contain any comparison among the variables in $M \cup N$. Consequently, using Lemma 3.9, for all linearization L_U for C_U , $L_U \models h(r) \rho h(s)$.

2- $x_1, x_2 \in N$. Similar to the case 1-.

3- $x_1 \in M, x_2 \in N$. Using the case 1- we can show that for $n \in M \cap N$, n and x_1 , and similarly n and x_2 are homomorphically equivalent wrt qv . Consequently, x_1 and x_2 are homomorphically equivalent.

4- $x_1 \in N, x_2 \in M$. Similar to the case 4-.

□

LEMMA 3.13. *Given a count-distinct query $q(\bar{x}, c(y)) :- A(\bar{x}, y, z) \wedge C(\bar{x}, y, z)$, and a variable w , $equiv(x, q)$ can be computed using Polynomial space in the size of the query.*

PROOF. Given a query q , we need to guess a subset of variables of that satisfies the conditions of an equal set, and check if there is no superset of this set that also satisfies the conditions of an equal set. This can be done in Σ_2^P . However, checking homomorphism itself is a Π_2^P problem. □

Now we are ready to define the notion of flip of a query.

DEFINITION 3.14 (FLIP A QUERY). *Given a count-distinct query q , the flip of q , denoted as $flip(q)$ is the count-distinct query obtained from changing the direction of all comparison operators of $equiv(y, q)$.*

Notice that by this definition the flip of the query will be equal to the query itself when $equiv(y, q) = \{y\}$.

LEMMA 3.15. *Given a count-distinct query $q_1(\bar{x}, c(y)) :- A(\bar{x}, y, z) \wedge C_1(\bar{x}, y, z)$, and its flip $q_2(\bar{x}, c(y)) :- A(\bar{x}, y, z) \wedge C_2(\bar{x}, y, z)$, $q_1 \equiv^\sigma q_2$, for $\sigma = \bar{x} \mapsto \bar{x}$.*

Notice that not necessarily $q_1 \equiv^{\sigma'} q_2$ for $\sigma = \bar{x}y \mapsto \bar{x}y$.

LEMMA 3.16 (EQUIVALENCE OF THE FLIP-SET QUERIES). *Consider the count-distinct query $q(\bar{x}) :- A(\bar{x}, \bar{y}) \wedge C(\bar{x}, \bar{y})$, and $q' = flip(q)$. $q = q'$.*

PROOF IDEA. By definition q' has been obtained from flipping the comparisons of the equal set of y . First notice that if $|equiv(y, q)| > 1$, then $equiv(y, q)$ cannot contain any variable from \bar{x} . For all variables x_1 and x_2 in the equivalence class $equiv(y, q)$, the existence of homomorphisms $\rightarrow_{\bar{x} \mapsto \bar{x}, x_1 \mapsto x_2}$ and $\rightarrow_{\bar{x} \mapsto \bar{x}, x_2 \mapsto x_1}$ guarantees that they match exactly the same set of terms for each group by value $\bar{x} \rightarrow \bar{d}$ in answering q' over all databases. Since $equiv(x, q)$ does not contain any variable from the group by variables, for the purpose of equivalence and containment of the queries it's only important to check for all assignments \bar{d} to \bar{x} they match exactly same number of items in q .

□

3.2.2 Equivalence of Count Distinct Queries

LEMMA 3.17. *Given a set of variables M , a variable y in M , a set C of comparisons among variables of M , and a set of ordered values D . Lets consider the DAG constructed out of C in which the variables are the nodes, and there is a and edge with weight 1 from x to y if $X > y$ and an edge with weight 0 if $x \geq 0$. The number of different values that can be assigned to y through assignments of values of D to the variables of M that satisfy all comparisons of C is $|D| - (M_h + M_l)$, in which M_h is the maximum distance of the variable y from one of its parents, and M_l is the maximum distance of y from one of its children in the DAG representing the comparisons of D .*

LEMMA 3.18. *Lets consider two sets of variables M and M' , variables $y \in M$ and $y' \in M'$, and set C (C') of comparisons among variables of M (M'). Lets consider the DAG D (D') constructed out of C (C') in which the variables are the nodes, and there is a and edge with weight 1 from x to y if $X > y$ and an edge with weight 0 if $x \geq 0$. With $ans(y, C, D)$ we denote the number of different assignments of values of D to the variables in C that satisfy all comparisons of C .*

For all set D of values from an ordered domain, $ans(y, C, D) \leq ans(y', C', D)$ if and only if one of followings hold:

- *There exists a homomorphism h from DAG of C' to DAG of C such that for all linearization L of the comparisons of C , for all comparison $r \rho s \in C'$, $L \models h(r)\rho h(s)$;*
- *Lets denote the set of comparisons obtained from C' by flipping their direction with C'' . There exists a homomorphism h'' from DAG of C'' to DAG of C such that for all linearization L of the comparisons of C , for all comparison $r \rho s \in C''$, $L \models h(r)\rho h(s)$;*

PROOF. Direct consequence of the Lemma 3.17. □

THEOREM 3.19. *Given two count distinct queries, it is decidable to check if they are equivalent.*

PROOF IDEA. The algorithm to check the equivalence of count-distinct queries is as follows. Lets consider the two queries to be compared are $q_1(\bar{x}, c(y)) :- A_1(\bar{x}, y, z) \wedge C_1(\bar{x}, y, z)$ and $q_2(\bar{x}, c(y)) :- A_2(\bar{x}, y, z) \wedge C_2(\bar{x}, y, z)$,

- Compute $q'_1 = core(q_1)$ and $q'_2 = core(q_2)$;
- Compute $q'_{1f} = flip(q_1)$ and $q'_{2f} = core(q_2)$;
- If for all $q''_1 \in \{q'_1, q'_{1f}\}$ and for all $q''_2 \in \{q'_2, q'_{2f}\}$ $q''_1 \not\equiv^{\bar{x}} q''_2$ then q_1 and q_2 are not equal; otherwise, they are equal.

We need to show that that the isomorphism between the core or flip of the cores of the queries is a sufficient and necessary condition for deciding their equivalence. Using Lemma 3.6, we know that the q_1 is equal to q'_1 and q_2 is equal to q'_2 .

Lemma 3.16, implies that q'_{1f} is equal to q'_1 and q'_{2f} is equal to q'_2 . As a result if one of q'_1 or q'_{1f} is isomorphic to one q'_2 or q'_{2f} then we can infer that q_1 and q_2 are also equal.

□

4. CONCLUSION

In this paper we discussed the problem of equivalence of count-distinct queries with comparisons, and proved the decidability of the problem through introducing two new notions of core of queries with comparisons and their flip over dense ordered domains. It is still open what will happen over discrete domains. Since we address the problem of equality through checking isomorphism, another legitimate open question is what will happen to the problem of containment of count-distinct queries.⁵

5. REFERENCES

- [1] S. Cohen. Containment of aggregate queries. *SIGMOD Record*, 34(1):77–85, 2005.
- [2] S. Cohen, W. Nutt, and Y. Sagiv. Deciding equivalences among conjunctive aggregate queries. *Journal of the ACM*, 54(2), 2007.
- [3] S. Cohen, Y. Sagiv, and W. Nutt. Equivalences among aggregate queries with negation. *ACM Transactions on Computational Logic*, 6(2):328–360, 2005.
- [4] A. C. Klug. On conjunctive queries containing inequalities. *Journal of the ACM*, 35(1):146–160, 1988.
- [5] W. Nutt, Y. Sagiv, and S. Shurin. Deciding equivalences among aggregate queries. In *Proceedings of the Seventeenth ACM SIGACT SIGMOD SIGART Symposium on Principles of Database Systems (PODS'98)*, pages 214–223, 1998.